# *What's Good About Them?*

## Some V&V Highlights from Past ASE Conferences

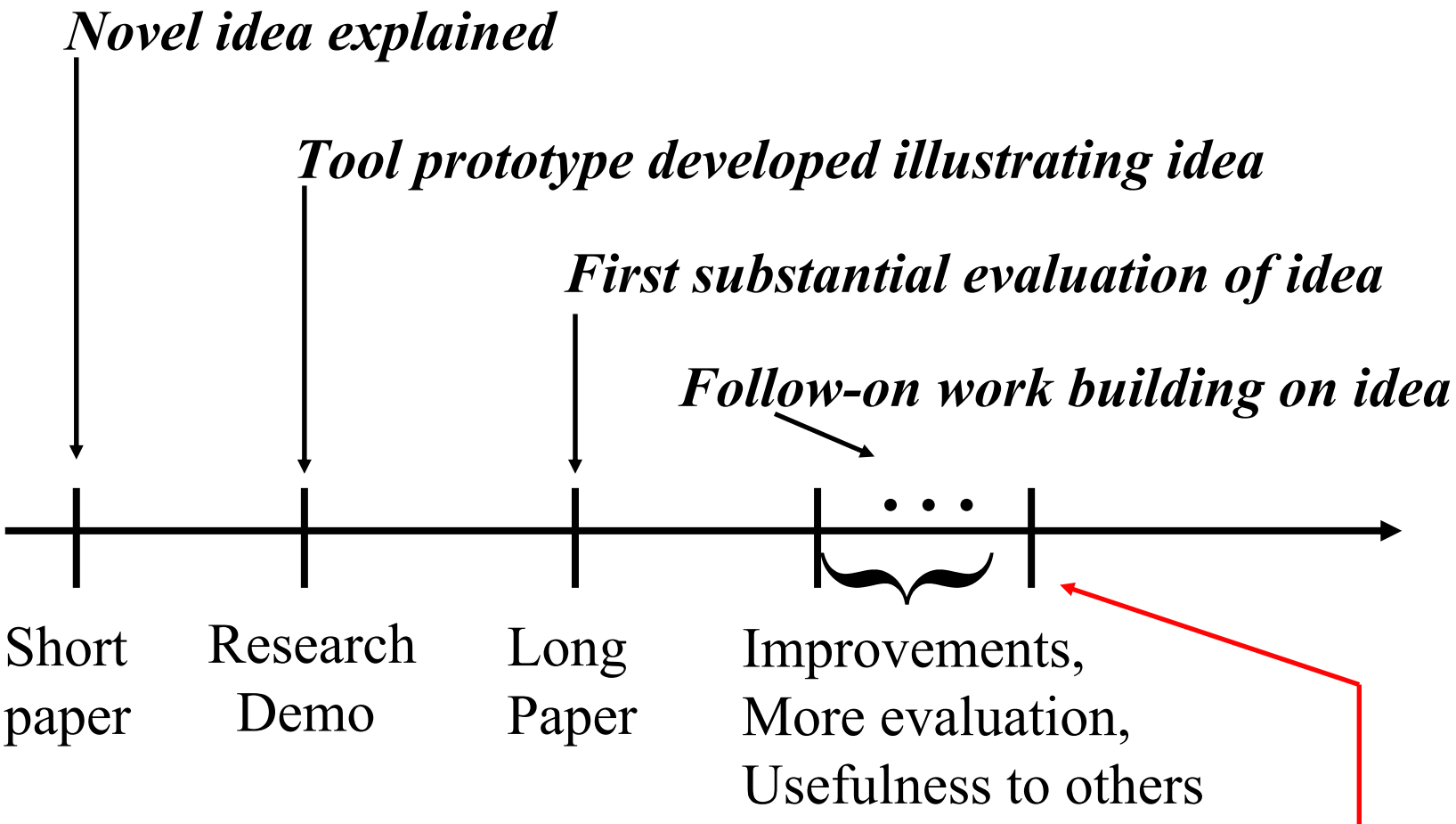
Dr. Robert J. Hall

AT&T Labs Research

# ASE: Beyond the Leading Edge

- ASE encourages early sharing of novel ideas
  - Short paper track, workshops
  - Facilitates discussion, leads to mature results later
- Tools: the "gold standard" at ASE
  - Research demonstrations track
  - Most ASE work is aimed at new tools
  - Working prototypes are influential
- ASE publishes convincing new results
  - Long paper track
  - Novel results backed by solid support: evaluation
  - Evaluation is often case studies involving tool prototype

# ASE as Incubator




**This talk: a few examples * that have progressed over here.**

**(* Not all were presented in all forms at ASE, of course.)**

# JPF Paper

- **"Model checking programs"** by Visser, Havelund, Brat, and Park, **ASE 2000**
- Idea: Move model checking from "modeling languages" into Java so it can be more easily applied to software
- Supported by ***Java Pathfinder*** model checker tool
- Substantial evaluation on avionics and spacecraft software in paper.
- Tool subsequently used widely by other groups for many applications

# Design for Verification

- "**Application of design for verification with concurrency controllers to air traffic control software"** by Betin-Can, Bultan, Lindvall, Lux, Topp, **ASE 2005**
- Main idea originally presented/evaluated in **ASE 2004** long paper by Betin-Can and Bultan. The 2005 paper presents a new, substantial evaluation of the idea.
- Idea: design concurrent applications using a particular design pattern that is constrained in such a way as to divide and conquer the verification problem: apply infinite state model checker to concurrency controller, Java Pathfinder to the threads themselves
- Second paper adds substantial weight to body of work

# Model Checking Product Lines

- "**Parameterized interfaces for open system verification of product lines"** by Blundell, Fisler, Krishnamurthi, Van Hentenryck, **ASE 2004**
- Idea: product family members are often assembled as new configurations of existing components; let's save time in verification by "precompiling" the model checking of each component in way allowing inexpensive checking at composition time.
- This paper is not only a substantial result, evaluated on a substantial case study, but it is also shown to be an improvement of their **ASE 2002** paper

# Systems Useful in the Real World

- "**Automatic generation of test oracles - from pilot studies to application**" by Feather and Smith, **ASE 1999**
- "**Upgrading legacy instances of reactive systems**" by Hall, **ASE 2000**
- "**Static consistency checking for distributed specifications**" by Nentwich, Emmerich, and Finkelstein, **ASE 2001**
- All have substantial evaluations validating systems (on which people depend) used in the real world (not synthetic data/bugs or post hoc reconstructions)